Deploying OpenShift Data Foundation using Microsoft Azure

Instructions on deploying OpenShift Data Foundation using Microsoft Azure
Instructions on deploying OpenShift Data Foundation using Microsoft Azure
Abstract

Read this document for instructions about how to install and manage Red Hat OpenShift Data Foundation using Red Hat OpenShift Container Platform on Microsoft Azure.
Table of Contents

**MAKING OPEN SOURCE MORE INCLUSIVE** ......................................................... 3
**PROVIDING FEEDBACK ON RED HAT DOCUMENTATION** .................................. 4
**PREFACE** ................................................................................................. 5

**CHAPTER 1. PREPARING TO DEPLOY OPENSIFICHT DATA FOUNDATION** .............. 6

**CHAPTER 2. DEPLOYING OPENSIFICHT DATA FOUNDATION ON MICROSOFT AZURE** .......... 7
  2.1. INSTALLING RED HAT OPENSIFICHT DATA FOUNDATION OPERATOR ............. 7
  2.2. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE TOKEN AUTHENTICATION METHOD ................................................................. 8
  2.3. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE KUBERNETES AUTHENTICATION METHOD ............................................................. 9
  2.4. CREATING AN OPENSIFICHT DATA FOUNDATION CLUSTER ....................... 12

**CHAPTER 3. VERIFYING OPENSIFICHT DATA FOUNDATION DEPLOYMENT** ............. 16
  3.1. VERIFYING THE STATE OF THE PODS ......................................................... 16
  3.2. VERIFYING THE OPENSIFICHT DATA FOUNDATION CLUSTER IS HEALTHY .......... 18
  3.3. VERIFYING THE MULTICLOUD OBJECT GATEWAY IS HEALTHY ................. 18
  3.4. VERIFYING THAT THE SPECIFIC STORAGE CLASSES EXIST ....................... 18

**CHAPTER 4. DEPLOY STANDALONE MULTICLOUD OBJECT GATEWAY** ..................... 19
  4.1. INSTALLING RED HAT OPENSIFICHT DATA FOUNDATION OPERATOR ........... 19
  4.2. CREATING A STANDALONE MULTICLOUD OBJECT GATEWAY .................... 20

**CHAPTER 5. VIEW OPENSIFICHT DATA FOUNDATION TOPOLOGY** ......................... 24

**CHAPTER 6. UNINSTALLING OPENSIFICHT DATA FOUNDATION** ............................ 25
  6.1. UNINSTALLING OPENSIFICHT DATA FOUNDATION IN INTERNAL MODE .......... 25
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Do let us know how we can make it better.

To give feedback, create a Bugzilla ticket:

1. Go to the Bugzilla website.

2. In the Component section, choose documentation.

3. Fill in the Description field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.

4. Click Submit Bug.
Red Hat OpenShift Data Foundation supports deployment on existing Red Hat OpenShift Container Platform (RHOC) Azure clusters.

**NOTE**

Only internal OpenShift Data Foundation clusters are supported on Microsoft Azure. See Planning your deployment for more information about deployment requirements.

To deploy OpenShift Data Foundation, start with the requirements in Preparing to deploy OpenShift Data Foundation chapter and then follow the appropriate deployment process based on your requirement:

- Deploy OpenShift Data Foundation on Microsoft Azure
- Deploy standalone Multicloud Object Gateway component
CHAPTER 1. PREPARING TO DEPLOY OPENSIGHT DATA FOUNDATION

Deploying OpenShift Data Foundation on OpenShift Container Platform using dynamic storage devices provides you with the option to create internal cluster resources. This will result in the internal provisioning of the base services, which helps to make additional storage classes available to applications.

Before you begin the deployment of OpenShift Data Foundation, follow these steps:

1. Setup a chrony server. See Configuring chrony time service and use knowledgebase solution to create rules allowing all traffic.

2. Optional: If you want to enable cluster-wide encryption using the external Key Management System (KMS) HashiCorp Vault, follow these steps:
   ● Ensure that you have a valid Red Hat OpenShift Data Foundation Advanced subscription. To know how subscriptions for OpenShift Data Foundation work, see knowledgebase article on OpenShift Data Foundation subscriptions.
   ● When the Token authentication method is selected for encryption then refer to Enabling cluster-wide encryption with the Token authentication using KMS.
   ● When the Kubernetes authentication method is selected for encryption then refer to Enabling cluster-wide encryption with the Kubernetes authentication using KMS.
   ● Ensure that you are using signed certificates on your Vault servers.

   **NOTE**
   If you are using Thales CipherTrust Manager as your KMS, you will enable it during deployment.

3. Minimum starting node requirements
   An OpenShift Data Foundation cluster is deployed with minimum configuration when the standard deployment resource requirement is not met. See Resource requirements section in Planning guide.

4. Disaster recovery requirements
   Disaster Recovery features supported by Red Hat OpenShift Data Foundation require all of the following prerequisites to successfully implement a disaster recovery solution:
   ● A valid Red Hat OpenShift Data Foundation Advanced subscription
   ● A valid Red Hat Advanced Cluster Management for Kubernetes subscription
     To know how subscriptions for OpenShift Data Foundation work, see knowledgebase article on OpenShift Data Foundation subscriptions.

For detailed requirements, see Configuring OpenShift Data Foundation Disaster Recovery for OpenShift Workloads guide, and Requirements and recommendations section of the Install guide in Red Hat Advanced Cluster Management for Kubernetes documentation.
CHAPTER 2. DEPLOYING OPENSHIFT DATA FOUNDATION ON MICROSOFT AZURE

You can deploy OpenShift Data Foundation on OpenShift Container Platform using dynamic storage devices provided by Microsoft Azure installer-provisioned infrastructure (IPI) (type: managed-csi) that enables you to create internal cluster resources. This results in internal provisioning of the base services, which helps to make additional storage classes available to applications.

Also, it is possible to deploy only the Multicloud Object Gateway (MCG) component with OpenShift Data Foundation. For more information, see Deploy standalone Multicloud Object Gateway.

NOTE

Only internal OpenShift Data Foundation clusters are supported on Microsoft Azure. See Planning your deployment for more information about deployment requirements.

Ensure that you have addressed the requirements in Preparing to deploy OpenShift Data Foundation chapter before proceeding with the below steps for deploying using dynamic storage devices:

1. Install the Red Hat OpenShift Data Foundation Operator.
2. Create the OpenShift Data Foundation Cluster

2.1. INSTALLING RED HAT OPENSHIFT DATA FOUNDATION OPERATOR

You can install Red Hat OpenShift Data Foundation Operator using the Red Hat OpenShift Container Platform Operator Hub.

Prerequisites

- Access to an OpenShift Container Platform cluster using an account with cluster-admin and operator installation permissions.
- You must have at least three worker or infrastructure nodes in the Red Hat OpenShift Container Platform cluster.
- For additional resource requirements, see the Planning your deployment guide.

IMPORTANT

- When you need to override the cluster-wide default node selector for OpenShift Data Foundation, you can use the following command to specify a blank node selector for the openshift-storage namespace (create openshift-storage namespace in this case):

```
$ oc annotate namespace openshift-storage openshift.io/node-selector=
```

- Taint a node as infra to ensure only Red Hat OpenShift Data Foundation resources are scheduled on that node. This helps you save on subscription costs. For more information, see the How to use dedicated worker nodes for Red Hat OpenShift Data Foundation section in the Managing and Allocating Storage Resources guide.
Procedure

1. Log in to the OpenShift Web Console.

2. Click Operators → OperatorHub.

3. Scroll or type OpenShift Data Foundation into the Filter by keyword box to find the OpenShift Data Foundation Operator.

4. Click Install.

5. Set the following options on the Install Operator page:
   b. Installation Mode as A specific namespace on the cluster
   c. Installed Namespace as Operator recommended namespace openshift-storage. If Namespace openshift-storage does not exist, it is created during the operator installation.
   d. Select Approval Strategy as Automatic or Manual.
      If you select Automatic updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.
      If you select Manual updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.
   e. Ensure that the Enable option is selected for the Console plugin.
   f. Click Install.

Verification steps

- After the operator is successfully installed, a pop-up with a message, Web console update is available appears on the user interface. Click Refresh web console from this pop-up for the console changes to reflect.

- In the Web Console:
  o Navigate to Installed Operators and verify that the OpenShift Data Foundation Operator shows a green tick indicating successful installation.
  o Navigate to Storage and verify if the Data Foundation dashboard is available.

2.2. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE TOKEN AUTHENTICATION METHOD

You can enable the key value backend path and policy in the vault for token authentication.

Prerequisites

- Administrator access to the vault.
- A valid Red Hat OpenShift Data Foundation Advanced subscription. For more information, see the knowledgebase article on OpenShift Data Foundation subscriptions.
Procedure

1. Enable the Key/Value (KV) backend path in the vault.
   For vault KV secret engine API, version 1:
   ```
   $ vault secrets enable -path=odf kv
   ```
   For vault KV secret engine API, version 2:
   ```
   $ vault secrets enable -path=odf kv-v2
   ```

2. Create a policy to restrict the users to perform a write or delete operation on the secret:
   ```
   echo 'path "odf/\*" {
   capabilities = ["create", "read", "update", "delete", "list"]
   }
   path "sys/mounts" {
   capabilities = ["read"]
   }' | vault policy write odf -
   ```

3. Create a token that matches the above policy:
   ```
   $ vault token create -policy=odf -format json
   ```

2.3. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE KUBERNETES AUTHENTICATION METHOD

You can enable the Kubernetes authentication method for cluster-wide encryption using the Key Management System (KMS).

Prerequisites

- Administrator access to Vault.
- A valid Red Hat OpenShift Data Foundation Advanced subscription. For more information, see the knowledgebase article on OpenShift Data Foundation subscriptions.
- The OpenShift Data Foundation operator must be installed from the Operator Hub.
- Select a unique path name as the backend path that follows the naming convention carefully. You cannot change this path name later.

Procedure

1. Create a service account:
   ```
   $ oc -n openshift-storage create serviceaccount <serviceaccount_name>
   ```
where, `<serviceaccount_name>` specifies the name of the service account.

For example:

```
$ oc -n openshift-storage create serviceaccount odf-vault-auth
```

2. Create **clusterrolebindings** and **clusterroles**:

```
$ oc -n openshift-storage create clusterrolebinding vault-tokenreview-binding --clusterrole=system:auth-delegator --serviceaccount=openshift-storage:_<serviceaccount_name>_
```

For example:

```
$ oc -n openshift-storage create clusterrolebinding vault-tokenreview-binding --clusterrole=system:auth-delegator --serviceaccount=openshift-storage:odf-vault-auth
```

3. Create a secret for the **serviceaccount** token and CA certificate.

```
$ cat <<EOF | oc create -f -
apiVersion: v1
kind: Secret
metadata:
  name: odf-vault-auth-token
  namespace: openshift-storage
annotations:
  kubernetes.io/service-account.name: <serviceaccount_name>
type: kubernetes.io/service-account-token
data: {}
EOF
```

where, `<serviceaccount_name>` is the service account created in the earlier step.

4. Get the token and the CA certificate from the secret.

```
$ SA_JWT_TOKEN=$(oc -n openshift-storage get secret odf-vault-auth-token -o jsonpath="{.data['token']}" | base64 --decode; echo)
$ SA_CA_CRT=$(oc -n openshift-storage get secret odf-vault-auth-token -o jsonpath="{.data['ca.crt']}" | base64 --decode; echo)
```

5. Retrieve the OCP cluster endpoint.

```
$ OCP_HOST=$(oc config view --minify --flatten -o jsonpath="{.clusters[0].cluster.server}")
```

6. Fetch the service account issuer:

```
$ oc proxy &
$ proxy_pid=$!
$ issuer="$( curl --silent http://127.0.0.1:8001/.well-known/openid-configuration | jq -r .issuer)"
$ kill $proxy_pid
```
7. Use the information collected in the previous step to set up the Kubernetes authentication method in Vault:

```bash
$ vault auth enable kubernetes
$ vault write auth/kubernetes/config
    token_reviewer_jwt="$SA_JWT_TOKEN"
    kubernetes_host="$OCP_HOST"
    kubernetes_ca_cert="$SA_CA_CRT"
    issuer="$issuer"
```

**IMPORTANT**

To configure the Kubernetes authentication method in Vault when the issuer is empty:

```bash
$ vault write auth/kubernetes/config
    token_reviewer_jwt="$SA_JWT_TOKEN"
    kubernetes_host="$OCP_HOST"
    kubernetes_ca_cert="$SA_CA_CRT"
```

8. Enable the Key/Value (KV) backend path in Vault.

For Vault KV secret engine API, version 1:

```bash
$ vault secrets enable -path=odf kv
```

For Vault KV secret engine API, version 2:

```bash
$ vault secrets enable -path=odf kv-v2
```

9. Create a policy to restrict the users to perform a **write** or **delete** operation on the secret:

```bash
echo 'path "odf/*" {
    capabilities = ["create", "read", "update", "delete", "list"]
} path "sys/mounts" {
    capabilities = ["read"]
}'} vault policy write odf -
```

10. Generate the roles:

```bash
$ vault write auth/kubernetes/role/odf-rook-ceph-op
    bound_service_account_names=rook-ceph-system,rook-ceph-osd,noobaa
    bound_service_account_namespaces=openshift-storage
    policies=odf
    ttl=1440h
```

The role **odf-rook-ceph-op** is later used while you configure the KMS connection details during the creation of the storage system.

```bash
$ vault write auth/kubernetes/role/odf-rook-ceph-osd
```

2.4. CREATING AN OPENSİFT DATA FOUNDATION CLUSTER

Create an OpenShift Data Foundation cluster after you install the OpenShift Data Foundation operator.

Prerequisites

- The OpenShift Data Foundation operator must be installed from the Operator Hub. For more information, see Installing OpenShift Data Foundation Operator using the Operator Hub.

Procedure

1. In the OpenShift Web Console, click Operators → Installed Operators to view all the installed operators. Ensure that the Project selected is openshift-storage.

2. Click on the OpenShift Data Foundation operator, and then click Create StorageSystem.

3. In the Backing storage page, select the following:
   a. Select Full Deployment for the Deployment type option.
   b. Select the Use an existing StorageClass option.
   c. Select the Storage Class.
      By default, it is set to managed-csi.
   d. Click Next.

4. In the Capacity and nodes page, provide the necessary information:
   a. Select a value for Requested Capacity from the dropdown list. It is set to 2 TiB by default.
      
      NOTE
      Once you select the initial storage capacity, cluster expansion is performed only using the selected usable capacity (three times of raw storage).

   b. In the Select Nodes section, select at least three available nodes.

   c. Optional: Select the Taint nodes checkbox to dedicate the selected nodes for OpenShift Data Foundation.
      For cloud platforms with multiple availability zones, ensure that the Nodes are spread across different Locations/availability zones.
      
      If the nodes selected do not match the OpenShift Data Foundation cluster requirements of an aggregated 30 CPUs and 72 GiB of RAM, a minimal cluster is deployed. For minimum starting node requirements, see the Resource requirements section in the Planning guide.
d. Optional [Technology preview]: Select the Add replica-1 pool checkbox to deploy OpenShift Data Foundation with a single replica. This avoids redundant data copies and allows resiliency management on the application level.

**WARNING**
Enabling this feature creates a single replica pool without data replication, increasing the risk of data loss, data corruption, and potential system instability if your application does not have its own replication.

**IMPORTANT**
Single replica deployment is a Technology Preview feature. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information, see Technology Preview Features Support Scope.

e. Click Next.

5. Optional: In the Security and network page, configure the following based on your requirements:

a. To enable encryption, select Enable data encryption for block and file storage

i. Select either one or both the encryption levels:

   - **Cluster-wide encryption**
     Encrypts the entire cluster (block and file).

   - **StorageClass encryption**
     Creates encrypted persistent volume (block only) using encryption enabled storage class.

ii. Optional: Select the Connect to an external key management service checkbox. This is optional for cluster-wide encryption.

   A. From the Key Management Service Provider drop-down list, either select Vault or Thales CipherTrust Manager (using KMIP). If you selected Vault, go to the next step. If you selected Thales CipherTrust Manager (using KMIP), go to step iii.

   B. Select an Authentication Method.

      **Using Token authentication method**

      - Enter a unique Connection Name, host Address of the Vault server (‘https://<hostname or ip>’), Port number and Token.
- Expand **Advanced Settings** to enter additional settings and certificate details based on your **Vault** configuration:
  - Enter the Key Value secret path in **Backend Path** that is dedicated and unique to OpenShift Data Foundation.
  - Optional: Enter **TLS Server Name** and **Vault Enterprise Namespace**.
  - Upload the respective PEM encoded certificate file to provide the **CA Certificate**, **Client Certificate** and **Client Private Key**.
  - Click **Save** and skip to step iv.

**Using Kubernetes authentication method**

- Enter a unique **Vault Connection Name**, host **Address** of the Vault server (`https://<hostname or ip>`), **Port** number and **Role** name.

- Expand **Advanced Settings** to enter additional settings and certificate details based on your **Vault** configuration:
  - Enter the Key Value secret path in **Backend Path** that is dedicated and unique to OpenShift Data Foundation.
  - Optional: Enter **TLS Server Name** and **Authentication Path** if applicable.
  - Upload the respective PEM encoded certificate file to provide the **CA Certificate**, **Client Certificate** and **Client Private Key**.
  - Click **Save** and skip to step iv.

C. To use **Thales CipherTrust Manager (using KMIP)** as the KMS provider, follow the steps below:

I. Enter a unique **Connection Name** for the Key Management service within the project.

II. In the **Address** and **Port** sections, enter the IP of Thales CipherTrust Manager and the port where the KMIP interface is enabled. For example:

   - **Address**: 123.34.3.2
   - **Port**: 5696

III. Upload the **Client Certificate**, **CA certificate**, and **Client Private Key**.

IV. If StorageClass encryption is enabled, enter the Unique Identifier to be used for encryption and decryption generated above.

V. The **TLS Server** field is optional and used when there is no DNS entry for the KMIP endpoint. For example, `kmip_all_<port>.ciphertrustmanager.local`.

D. Select a **Network**.

E. Click **Next**.
b. To enable in-transit encryption, select **In-transit encryption**.
   
   i. Select a **Network**.
   
   ii. Click **Next**.

6. In the **Review and create** page, review the configuration details. To modify any configuration settings, click **Back**.

7. Click **Create StorageSystem**.

**Verification steps**

- To verify the final Status of the installed storage cluster:
  
  a. In the OpenShift Web Console, navigate to **Installed Operators → OpenShift Data Foundation → Storage System → ocs-storagecluster-storagesystem → Resources**.
  
  b. Verify that **Status** of **StorageCluster** is **Ready** and has a green tick mark next to it.

- To verify that all components for OpenShift Data Foundation are successfully installed, see **Verifying your OpenShift Data Foundation deployment**.

**Additional resources**

To enable Overprovision Control alerts, refer to **Alerts** in Monitoring guide.
CHAPTER 3. VERIFYING OPENSHIFT DATA FOUNDATION DEPLOYMENT

Use this section to verify that OpenShift Data Foundation is deployed correctly.

3.1. VERIFYING THE STATE OF THE PODS

Procedure

1. Click Workloads → Pods from the OpenShift Web Console.

2. Select openshift-storage from the Project drop-down list.

   NOTE

   If the Show default projects option is disabled, use the toggle button to list all
   the default projects.

   For more information on the expected number of pods for each component and how it varies
   depending on the number of nodes, see Table 3.1, “Pods corresponding to OpenShift Data
   Foundation cluster”.

3. Set filter for Running and Completed pods to verify that the following pods are in Running and
   Completed state:

   Table 3.1. Pods corresponding to OpenShift Data Foundation cluster

<table>
<thead>
<tr>
<th>Component</th>
<th>Corresponding pods</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenShift Data Foundation Operator</td>
<td>• ocs-operator-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>• ocs-metrics-exporter-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>• odf-operator-controller-manager-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>• odf-console-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>• csi-addons-controller-manager-* (1 pod on any storage node)</td>
</tr>
<tr>
<td>Rook-ceph Operator</td>
<td>rook-ceph-operator-*</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
<tr>
<td>Component</td>
<td>Corresponding pods</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Multicloud Object Gateway</td>
<td>- noobaa-operator-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>- noobaa-core-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>- noobaa-db-pg-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>- noobaa-endpoint-* (1 pod on any storage node)</td>
</tr>
<tr>
<td>MON</td>
<td>rook-ceph-mon-*</td>
</tr>
<tr>
<td></td>
<td>(3 pods distributed across storage nodes)</td>
</tr>
<tr>
<td>MGR</td>
<td>rook-ceph-mgr-*</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
<tr>
<td>MDS</td>
<td>rook-ceph-mds-ocs-storagecluster-cephfilesystem-*</td>
</tr>
<tr>
<td></td>
<td>(2 pods distributed across storage nodes)</td>
</tr>
<tr>
<td>CSI</td>
<td>- cephfs</td>
</tr>
<tr>
<td></td>
<td>- csi-cephfsplugin-* (1 pod on each storage node)</td>
</tr>
<tr>
<td></td>
<td>- csi-cephfsplugin-provisioner-* (2 pods distributed across storage nodes)</td>
</tr>
<tr>
<td></td>
<td>- rbd</td>
</tr>
<tr>
<td></td>
<td>- csi-rbdplugin-* (1 pod on each storage node)</td>
</tr>
<tr>
<td></td>
<td>- csi-rbdplugin-provisioner-* (2 pods distributed across storage nodes)</td>
</tr>
<tr>
<td></td>
<td>rook-ceph-crashcollector-*</td>
</tr>
<tr>
<td></td>
<td>(1 pod on each storage node)</td>
</tr>
<tr>
<td>OSD</td>
<td>- rook-ceph-osd-*</td>
</tr>
<tr>
<td></td>
<td>- rook-ceph-osd-prepare-ocs-deviceset-* (1 pod for each device)</td>
</tr>
</tbody>
</table>
3.2. VERIFYING THE OPENSHIFT DATA FOUNDATION CLUSTER IS HEALTHY

Procedure

1. In the OpenShift Web Console, click **Storage → Data Foundation**.

2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.

3. In the **Status** card of the **Block and File** tab, verify that the **Storage Cluster** has a green tick.

4. In the **Details** card, verify that the cluster information is displayed.

For more information on the health of the OpenShift Data Foundation cluster using the **Block and File** dashboard, see Monitoring OpenShift Data Foundation.

3.3. VERIFYING THE MULTICLOUD OBJECT GATEWAY IS HEALTHY

Procedure

1. In the OpenShift Web Console, click **Storage → Data Foundation**.

2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.

   a. In the **Status card** of the **Object** tab, verify that both **Object Service** and **Data Resiliency** have a green tick.

   b. In the **Details** card, verify that the MCG information is displayed.

For more information on the health of the OpenShift Data Foundation cluster using the object service dashboard, see Monitoring OpenShift Data Foundation.

3.4. VERIFYING THAT THE SPECIFIC STORAGE CLASSES EXIST

Procedure

1. Click **Storage → Storage Classes** from the left pane of the OpenShift Web Console.

2. Verify that the following storage classes are created with the OpenShift Data Foundation cluster creation:

   - ocs-storagecluster-ceph-rbd
   - ocs-storagecluster-cephfs
   - openshift-storage.noobaa.io
Deploying only the Multicloud Object Gateway component with OpenShift Data Foundation provides the flexibility in deployment and helps to reduce the resource consumption. Use this section to deploy only the standalone Multicloud Object Gateway component, which involves the following steps:

- Installing Red Hat OpenShift Data Foundation Operator
- Creating standalone Multicloud Object Gateway

### 4.1. INSTALLING RED HAT OPENSIFHT DATA FOUNDATION OPERATOR

You can install Red Hat OpenShift Data Foundation Operator using the Red Hat OpenShift Container Platform Operator Hub.

#### Prerequisites

- Access to an OpenShift Container Platform cluster using an account with `cluster-admin` and operator installation permissions.
- You must have at least three worker or infrastructure nodes in the Red Hat OpenShift Container Platform cluster.
- For additional resource requirements, see the Planning your deployment guide.

#### IMPORTANT

- When you need to override the cluster-wide default node selector for OpenShift Data Foundation, you can use the following command to specify a blank node selector for the `openshift-storage` namespace (create `openshift-storage` namespace in this case):

  ```
  $ oc annotate namespace openshift-storage openshift.io/node-selector= 
  ```

- Taint a node as `infra` to ensure only Red Hat OpenShift Data Foundation resources are scheduled on that node. This helps you save on subscription costs. For more information, see the How to use dedicated worker nodes for Red Hat OpenShift Data Foundation section in the Managing and Allocating Storage Resources guide.

#### Procedure

1. Log in to the OpenShift Web Console.
2. Click Operators → OperatorHub.
3. Scroll or type OpenShift Data Foundation into the Filter by keyword box to find the OpenShift Data Foundation Operator.
4. Click Install.
5. Set the following options on the Install Operator page:
a. Update Channel as **stable-4.14**.

b. Installation Mode as **A specific namespace on the cluster**

c. Installed Namespace as **Operator recommended namespace openshift-storage**. If Namespace openshift-storage does not exist, it is created during the operator installation.

d. Select Approval Strategy as **Automatic** or **Manual**.
   If you select **Automatic** updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.
   
   If you select **Manual** updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.

e. Ensure that the **Enable** option is selected for the **Console plugin**.

f. Click **Install**.

**Verification steps**

- After the operator is successfully installed, a pop-up with a message, **Web console update is available** appears on the user interface. Click **Refresh web console** from this pop-up for the console changes to reflect.

- In the Web Console:
  - Navigate to Installed Operators and verify that the **OpenShift Data Foundation** Operator shows a green tick indicating successful installation.
  - Navigate to **Storage** and verify if the **Data Foundation** dashboard is available.

4.2. CREATING A STANDALONE MULTICLOUD OBJECT GATEWAY

You can create only the standalone Multicloud Object Gateway component while deploying OpenShift Data Foundation.

**Prerequisites**

- Ensure that the OpenShift Data Foundation Operator is installed.

**Procedure**

1. In the OpenShift Web Console, click **Operators → Installed Operators** to view all the installed operators.
   Ensure that the **Project** selected is openshift-storage.

2. Click **OpenShift Data Foundation** operator and then click **Create StorageSystem**.

3. In the **Backing storage** page, select the following:
   a. Select **Multicloud Object Gateway** for **Deployment type**
   b. Select the **Use an existing StorageClass** option.
   c. Click **Next**.
4. Optional: Select the Connect to an external key management service checkbox. This is optional for cluster-wide encryption.

a. From the Key Management Service Provider drop-down list, either select Vault or Thales CipherTrust Manager (using KMIP). If you selected Vault, go to the next step. If you selected Thales CipherTrust Manager (using KMIP), go to step iii.

b. Select an Authentication Method.

Using Token authentication method

- Enter a unique Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number and Token.

- Expand Advanced Settings to enter additional settings and certificate details based on your Vault configuration:
  
  o Enter the Key Value secret path in Backend Path that is dedicated and unique to OpenShift Data Foundation.

  o Optional: Enter TLS Server Name and Vault Enterprise Namespace

  o Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate and Client Private Key.

  o Click Save and skip to step iv.

Using Kubernetes authentication method

- Enter a unique Vault Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number and Role name.

- Expand Advanced Settings to enter additional settings and certificate details based on your Vault configuration:

  o Enter the Key Value secret path in Backend Path that is dedicated and unique to OpenShift Data Foundation.

  o Optional: Enter TLS Server Name and Authentication Path if applicable.

  o Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate and Client Private Key.

  o Click Save and skip to step iv.

c. To use Thales CipherTrust Manager (using KMIP) as the KMS provider, follow the steps below:

i. Enter a unique Connection Name for the Key Management service within the project.

ii. In the Address and Port sections, enter the IP of Thales CipherTrust Manager and the port where the KMIP interface is enabled. For example:

  o Address: 123.34.3.2

  o Port: 5696

iii. Upload the Client Certificate, CA certificate, and Client Private Key.
iv. If StorageClass encryption is enabled, enter the Unique Identifier to be used for encryption and decryption generated above.

v. The **TLS Server** field is optional and used when there is no DNS entry for the KMIP endpoint. For example, `kmip_all_<port>.ciphertrustmanager.local`.

d. Select a **Network**.

e. Click **Next**.

5. In the **Review and create** page, review the configuration details:
   To modify any configuration settings, click **Back**.

6. Click **Create StorageSystem**.

**Verification steps**

**Verifying that the OpenShift Data Foundation cluster is healthy**

1. In the OpenShift Web Console, click **Storage → Data Foundation**.

2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.

   a. In the **Status card** of the **Object** tab, verify that both **Object Service** and **Data Resiliency** have a green tick.

   b. In the **Details** card, verify that the MCG information is displayed.

**Verifying the state of the pods**

1. Click **Workloads → Pods** from the OpenShift Web Console.

2. Select **openshift-storage** from the **Project** drop-down list and verify that the following pods are in **Running** state.

   **NOTE**
   If the **Show default projects** option is disabled, use the toggle button to list all the default projects.

<table>
<thead>
<tr>
<th>Component</th>
<th>Corresponding pods</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenShift Data Foundation Operator</td>
<td>- <em><em>ocs-operator-</em> (1 pod on any storage node)</em>*</td>
</tr>
<tr>
<td></td>
<td>- *<em>ocs-metrics-exporter-</em> (1 pod on any storage node)**</td>
</tr>
<tr>
<td></td>
<td>- *<em>odf-operator-controller-manager-</em> (1 pod on any storage node)**</td>
</tr>
<tr>
<td></td>
<td>- <em><em>odf-console-</em> (1 pod on any storage node)</em>*</td>
</tr>
<tr>
<td></td>
<td>- <em><em>csi-addons-controller-manager-</em> (1 pod on any storage node)</em>*</td>
</tr>
<tr>
<td>Component</td>
<td>Corresponding pods</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Rook-ceph Operator</td>
<td>rook-ceph-operator-*</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
<tr>
<td>Multicloud Object Gateway</td>
<td>● noobaa-operator-*</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>● noobaa-core-*</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>● noobaa-db-pg-*</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>● noobaa-endpoint-*</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
</tbody>
</table>
CHAPTER 5. VIEW OPENSIGHT DATA FOUNDATION TOPOLOGY

The topology shows the mapped visualization of the OpenShift Data Foundation storage cluster at various abstraction levels and also lets you to interact with these layers. The view also shows how the various elements compose the Storage cluster altogether.

Procedure

1. On the OpenShift Web Console, navigate to **Storage → Data Foundation → Topology**. The view shows the storage cluster and the zones inside it. You can see the nodes depicted by circular entities within the zones, which are indicated by dotted lines. The label of each item or resource contains basic information such as status and health or indication for alerts.

2. Choose a node to view node details on the right-hand panel. You can also access resources or deployments within a node by clicking on the search/preview decorator icon.

3. To view deployment details
   a. Click the preview decorator on a node. A modal window appears above the node that displays all of the deployments associated with that node along with their statuses.
   b. Click the **Back to main view** button in the model’s upper left corner to close and return to the previous view.
   c. Select a specific deployment to see more information about it. All relevant data is shown in the side panel.

4. Click the **Resources** tab to view the pods information. This tab provides a deeper understanding of the problems and offers granularity that aids in better troubleshooting.

5. Click the pod links to view the pod information page on OpenShift Container Platform. The link opens in a new window.
CHAPTER 6. UNINSTALLING OPENSSHIFT DATA FOUNDATION

6.1. UNINSTALLING OPENShift DATA FOUNDATION IN INTERNAL MODE

To uninstall OpenShift Data Foundation in Internal mode, refer to the knowledge base article on Uninstalling OpenShift Data Foundation.